Installation and Usage Instructions

The source code is written in Python and requires no special packages. Therefore, there is no special installation instructions. Only installing common python packages through standard package management systems, such as apt, yum and pip, is required. For information on the file, please check README.md

1. Operating Systems

The source code is tested on Debian 9 and Mac OSX 10.15. To run on Windows, please change the path delimiter from "/" to "\\".

2. Python Environment

   Ptyhon 3.7, from the referenced CPython implementation.

3. Required Python Packages
   Please install the following python packages:
   1) Numpy
   2) SciPy
   3) Matplotlib
   4) argparse

4. Usage
   1) Please put all python scripts and the data in the same directory. That is, the structure of the directory should be:

```
.
|-- PT4CloudArtifacts.py
|-- load_data.py
|-- pt4cloud_figures.py
|-- AWS   –  "AWS data directory"
|-- Chameleon – "Chameleon data directory"
```

   2) The code is used to generate the data in Evaluation section (Section 5), which are the evaluation results of PT4Cloud. More specifically, the results obtained from PT4Cloud in Table 3, Figure 4 and Figure 5.

   3) The results are partition based on benchmark configurations, including benchmark name, cloud platform and VM type. When invoking the scripts, the exact configuration information must be specified. The following table gives the configuration information for all benchmarks, platforms and VM types. For reduce method, please refer to the paper, we provide three configurable parameters "*no, yes, both*"

Config	Benchmark	Platform	VM Type	VM Config Name
1	ft	chameleon	m	CHM-2
2	ft	chameleon	l	CHM-3
3	ep	chameleon	m	CHM-2
4	ep	chameleon	l	CHM-3
5	jps	chameleon	s	CHM-1

6	jps	chameleon	m	CHM-2
7	jps	chameleon	l	CHM-3
8	ycsb	chameleon	s	CHM-1
9	ycsb	chameleon	m	CHM-2
10	ycsb	chameleon	l	CHM-3
11	tpcc	chameleon	s	CHM-1
12	tpcc	chameleon	m	CHM-2
13	tpcc	chameleon	l	CHM-3
14	ima	chameleon	m	CHM-2
15	ima	chameleon	l	CHM-3
16	ft	aws	l	AWS-1
17	ft	aws	xl	AWS-2
18	ft	aws	xxl	AWS-3
19	ep	aws	l	AWS-1
20	ep	aws	xl	AWS-2
21	ep	aws	xxl	AWS-3
22	jps	aws	l	AWS-1
23	jps	aws	xl	AWS-2
24	jps	aws	xxl	AWS-3
25	ycsb	aws	l	AWS-1
26	ycsb	aws	xl	AWS-2
27	ycsb	aws	xxl	AWS-3
28	tpcc	aws	l	AWS-1
29	tpcc	aws	xl	AWS-2
30	tpcc	aws	xxl	AWS-3
31	ima	aws	l	AWS-1
32	ima	aws	xl	AWS-2
33	ima	aws	xxl	AWS-3

Note that, for Chameleon, VM types “s”, “m” and “l” correspond to the CHM-1, CHM-2 and CHM-3 in Table 2. For AWS, VM types “l”, “xl” and “xxl” correspond to the AWS-1, AWS-2 and AWS-3 in Table 2.

4) To obtain the results, please invoke the main python script with the following format:

```
$ python3 PT4CloudArtifacts.py -b Benchmark -p Platform -v VMType -r ReduceMehod
```

Substitute the *benchmark*, *platform, VMType* and *ReduceMethod* with the information from the above table. For example, to generate the results for benchmark JpetStore on AWS with VM type m5.xl (AWS-2) with basic PT4Cloud, the script should be invoked as,

```
$ python3 PT4CloudArtifacts.py -b jps -p aws -v xl -r no
```

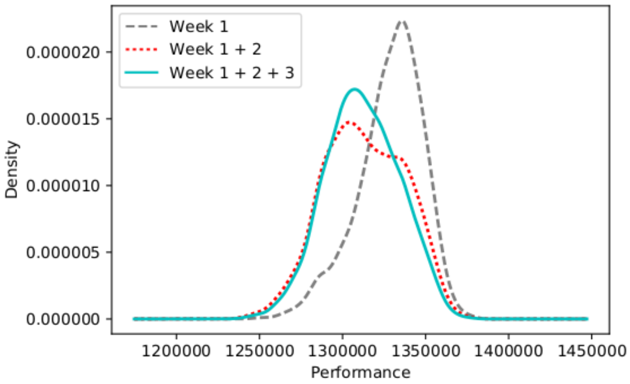
5) The output from the script will give the accuracy of PT4Cloud methodology shown in Table 3. For example, for the above invocation with JPS on AWS-2, the outputs are

```
Calculating the similarity of first 1 week(s) and the first 2 weeks
Similarity probability is 59.15%
Performance distribution is not stable, week 3 will be tested
Calculating the similarity of first 2 week(s) and the first 3 weeks
Similarity probability is 97.06%
Performance distribution is stable in 3 weeks
Ground truth validation succeed with probability 96.05%

Generating distribution figure for PT4Cloud testing results...

Generating validation figure for PT4Cloud testing results...
Bootstrapping confidence band, this may take a few minutes
```

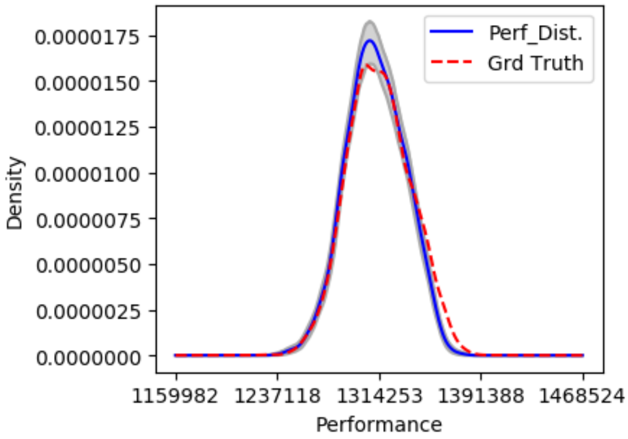
The highlighted “in 3 weeks” and 96.05% is the PT4Cloud test length and accuracy for JPS on AWS-2, which match the values of the cell at row 5 and column 6 of Table 3. Similarly, for all benchmark configurations, the corresponding testing length and accuracy can be generated using the supplied script (i.e., PT4CloudArtifacts.py).

6) Besides testing length and accuracy, the script will also generate a figure of the performance distribution matching those in Figure 4, and generate the validation figure (with ground truth and confidence band) as shown in Figure 5. Both figures will also be saved to the disk using benchmark name, platform name and VM type as file names.

For example, for the above example of JPS on AWS-2, the first figure (as shown below) matches Figure 4(f),



The second figure (as shown below) matches Figure 5(f).



Note that, limitation by space, Figure 4 and 5 in the paper do not include all benchmark configurations. However, our supplied can generate figures for all benchmarks.